Touch Detection with Low-cost Visual-based Sensor

Julio Castaño-Amoros[a], Pablo Gil[b] and Santiago Puente[c]
AUROVA Lab, Computer Science Research Institute, University of Alicante, Alicante 03690, Spain

Keywords: Tactile Sensing, Robotic Grasping, DIGIT Sensor, Convolutional Neural Networks.

Abstract: Robotic manipulation continues being an unsolved problem. It involves many complex aspects, for example, perception tactile of different objects and materials, grasping control to plan the robotic hand pose, etc. Most of previous works on this topic used expensive sensors. This fact makes difficult the application in the industry. In this work, we propose a grip detection system using a low-cost visual-based tactile sensor known as DIGIT, mounted on a ROBOTIQ gripper 2F-140. We proved that a Deep Convolutional Network is able to detect contact or no contact. Capturing almost 12000 images with contact and no contact from different objects, we achieve 99% accuracy with never seen samples, in the best scenario. As a result, this system will allow us to implement a grasping controller for the gripper.

1 INTRODUCTION

Tactile perception is becoming more and more essential in robotic manipulation tasks as shown in (Kappassov et al., 2015) and (Li et al., 2020). Touch data are often used to obtain information about manipulated objects such as shape, rigidity or texture of the material (Luo et al., 2017). This can be used for object recognition tasks, when visual information from eye-to-hand systems is not sufficient for a successful recognition.

Besides, touch data can be used to adapt the object grasping by correcting the opening or closing of the robotic hand or gripper (Delgado Rodríguez et al., 2017). Sometimes, it is used to plan finger movements in order to ensure a better grasp (Calandra et al., 2018), avoiding slipping or falling of the manipulated objects.

With the aim of controlling the gripper opening, or making a stable grasp with a robotic hand, state of art works showed methods based on machine learning techniques (Cockbum et al., 2017) and (Bekiroglu et al., 2011). Later, other methods used deep learning techniques (Kwiatkowski et al., 2017) and (Ni et al., 2019). Both types of approaches rely on the tactile sensor technology. A review of tactile technologies can be found in (Yi et al., 2018). They can be piezoresistive, capacitive, optical, magnetic, with barometric transducers, etc. Therefore, the neural architectures proposed to solve this problem are very different. For example, in (Zapata-Impata et al., 2018), a simple Convolutional Neural Network (CNN) is designed to detect stability in a virtual tactile image created from the electrode values of a BioTac sensor SP. This virtual tactile image was used to represent the connectivity among neighboring electrodes. In (Garcia-Garcia et al., 2019), a Graph Convolutional Network (GCN) was created to avoid building an intermediate representation like a virtual tactile image. GCN also keeps the connectivity among neighbors. The results of both methodologies were compared in (Zapata-Impata et al., 2019).

In this work, we present a first approach for the detection of tactile contact between the robot and object. We plan to use it as a part of a controller that allows the robot to manipulate an object in-hand without slipping. To do this, we used a novel low-cost visual-based tactile sensor known as DIGIT (Lambeta et al., 2020), instead of BioTac SP. DIGIT is based on an optical image whereas BioTac SP is based on barometric transducers, and is much more expensive.

This paper is organised as follows: first, we will describe the robotic grasping system consisting of a gripper with DIGIT tactile sensors mounted on the parallel fingertips. Second, we will present our methodology for contact detection with DIGIT sensor and artificial intelligence. Finally, we will show our results and conclusions, as well as future works.

[a] https://orcid.org/0000-0001-9789-1628
[b] https://orcid.org/0000-0001-9288-0161
[c] https://orcid.org/0000-0002-6175-600X

Castaño-Amoros, J., Gil, P. and Puente, S.
Touch Detection with Low-cost Visual-based Sensor.
DOI: 10.5220/0010699800003061
In *Proceedings of the 2nd International Conference on Robotics, Computer Vision and Intelligent Systems (ROBOVIS 2021)*, pages 136-142
ISBN: 978-989-758-537-1

2 ROBOTIC GRASPING SYSTEM

2.1 Visual-based Tactile Sensor

Robots need touch sensing to achieve human manipulation. In recent years, different types of tactile sensors have been used for this purpose. A game changer in robotic perception are tactile sensors based on visual technology such as Gelsight (Yuan et al., 2017), Gelslim (Donlon et al., 2018) or DIGIT. DIGIT is a sensor whose physical structure consists of an elastomer (which can be transparent, reflective or with markers), an acrylic window, a PCB camera, a PCB lighting, and a housing of several pieces that can be easily manufactured with a 3D printer. The camera captures up to 60 fps of colored images with a resolution of 240x320 pixels. Contact with the DIGIT implies deformation of the elastomer. This shape modification changes the way light travels from the elastomer to the camera. And so the image will be different as seen in Figure 3.

In this work, we built three units of DIGIT sensor (A, B and C). All the sensors were built with reflective elastomers without markers. Since sensors assembling is manual, it is almost impossible to apply the same quantity of paint and hardness in their manufacturing, which influences the elastomer's deformation. Besides, PETG material was used to print the housing.

2.2 Gripper System

We mounted a DIGIT sensor on each fingertip of a ROBOTIQ gripper 2F-140. This gripper has two articulated fingers with two joints on each one of them. The fingers are under-actuated. Therefore, they can adapt to the object's shape during grasping tasks. The gripper can do internal and external grasps. As we mounted the sensors in the internal face of the fingertips, our gripper performs external grasping. Otherwise, the grasping would be internal.

We designed and built an additional piece, in PETG material, to attach each sensor to each fingertip. Both devices, gripper and sensors, are connected using a serial communication protocol to a PC and controlled with Robot Operative System (ROS) Kinetic Kame. The gripper can be controlled in position, velocity and force. The gripper's operation mode allows us to detect contact between an object and the fingers. This mode works measuring the amperage, and comparing it with a pre-recorded reference. Nonetheless, we discarded this mode because each object has a different reference amperage. Then, measuring a new reference amperage value is needed every time we use a new object. In contrast, using DIGIT sensors, different objects produce similar shapes during the contact, hence, it is easier to generalise when using new objects. In this paper, we present a contact detection method based on the tactile image provided by DIGIT sensors.

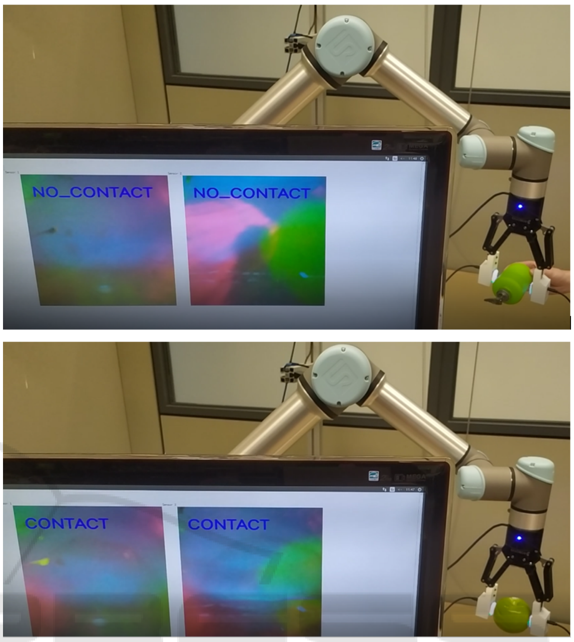

Figure 1: Gripper system.

3 SUPERVISED LEARNING METHODOLOGY

In this work, we did several experiments with classical computer vision and machine learning techniques, for example, finding contours in the contact area, detecting feature's movement using Optical Flow, or dimensionality reduction by means of Principal Component Analysis. However, DIGIT images are very noisy due to the merge of colours. We had to discard these methods because they could not extract good features from the images, and the results were not robust.

As our objective was to achieve contact detection with a wide panel of household objects, we chose to focus on deep learning methods. Indeed, we knew that those methods were capable of learning from image features. Working with images, we decided to focus on CNNs. They are capable of extracting features thanks to convolutional method and filters applied through different layers.

3.1 Neural Architecture

We are facing a binary classification task: contact or no contact between an object and the sensor. The neural architecture is composed of a backbone for feature extraction, fully connected layers, a final classifier with one neuron and a sigmoid as activation function.

To find the optimal backbone that allows us to detect tactile contact, we use three different models: VGG19 (Simonyan and Zisserman, 2014), InceptionV3 (Szegedy et al., 2016) and MobileNetV2 (Sandler et al., 2018). Using every existing backbone is impossible, so we analysed the performance of three models with different characteristics such as depth, inference, training time, size, etc.

VGG19 architecture contains only 19 layers with 3x3 standard convolution filters, max pooling layers, and a pair of fully connected layers at the end.

Unlike VGG19, InceptionV3 has a more complex architecture. The most important part of its architecture is the use of inception blocks, which apply convolutions changing the kernel's size to extract features with different levels of detail.

MobileNetV2 was designed to be executed in mobile devices. Thus, the authors created a model with as fewer parameters and mathematical operations as possible. This CNN shows the good performance of depthwise separable convolutions in accuracy and efficiency.

To evaluate the performance, we also applied a batch normalisation process as well as dropout technique, except for MobileNetV2, in which we only create the classifier.

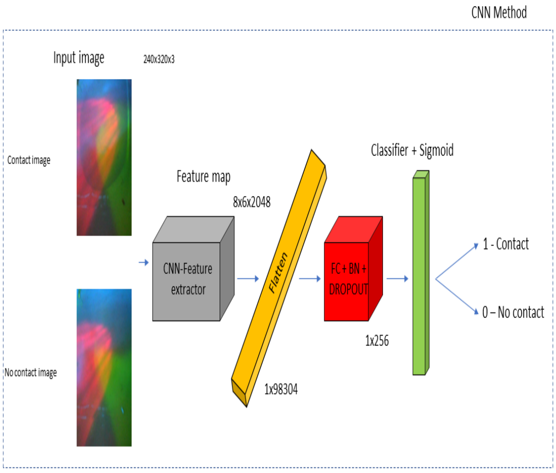

Figure 2: Neural architecture.

3.2 Tactile Data Collection

We assembled three DIGIT sensors following the instructions shown in (Lambeta et al., 2020). Each sensor provides a slightly different output image as shown in Figure 3, due to a manual assembly. In this article, we made a dataset with images from each sensor, and we trained each model described in aforementioned section.

To create the dataset, we used different objects such as shoe soles, air freshener, tennis ball, shampoo bottle, chips can, etc, (see Figure 4). This dataset is made up of two classes: contact and no contact. Both classes have approximately the same number of samples: 8200. Contact class contains 40% images from sensor A, 30% images from sensor B and 30% from sensor C, whilst no contact class contains 43% images from sensor A, 25% images from sensor B and 32% from sensor C. The number of samples from sensor A is higher because its gel has difficulties recovering its initial shape and colours after the contact. Background colours change more often in this sensor. Therefore, adding as many background images as possible from this sensor will help the model to learn correctly.

To make the model more robust, we need to record contact data varying forces and shapes. Hence, the model will be able to detect any kind of contact. Besides, recording data when the object is releasing the sensor and the gel is recovering its initial shape increases model's robustness.

4 EXPERIMENTATION

In this paper, we used a NVIDIA DGX A100 platform for training. All the experimentation is coded in Python 3.7.2, Keras 2.4.0 and Tensorflow 2.4.1. For inference, we used an i5-8400 CPU @ 2.8Ghz, 16 GIB DDR4 RAM.

4.1 Training Methodology

We performed a hyperparametric fine-tunning throughout the training phase of the neural architectures. Thus, we could choose a set of optimal hyperparameters for each CNN.

Note that, we split our data into three subsets: train (70%), validation (20%) and test (10%). Data were randomly chosen, ensuring that there were no identical samples in training, validation and test. It was not necessary to apply any kind of cross validation.

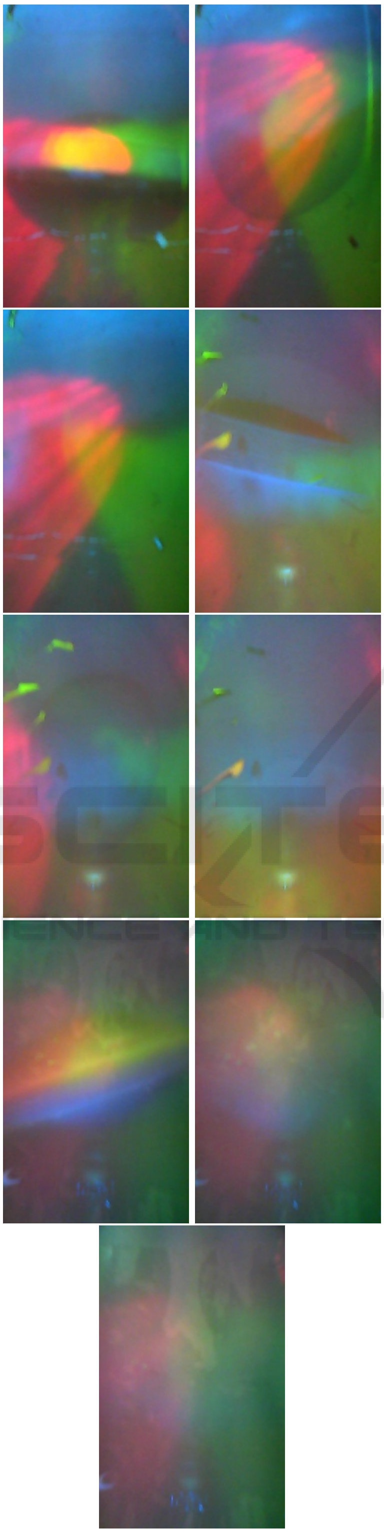

Figure 3: Sensor A, B and C (top to bottom), contact image (shoe sole), contact image (plastic apple) and no contact image (left to right).

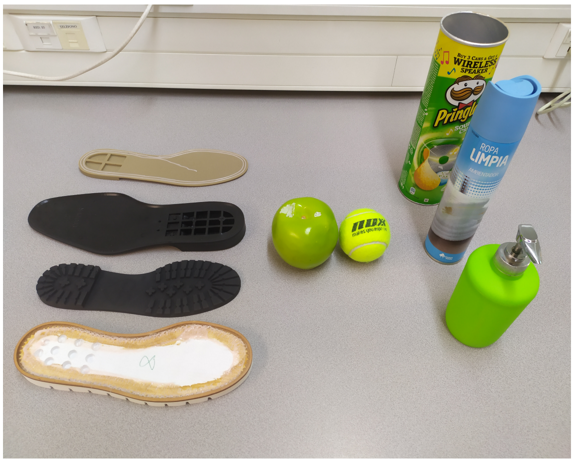

Figure 4: Objects used to create the dataset.

We applied data augmentation to increase the size of data used for successful tunning of the training models. To do it, in particular, we used different transformations of input image, such as 0.2 zoom range, 5° rotation range and horizontal flip.

Starting with VGG19 backbone, this model was trained several times to verify which approach is better among training from scratch, with transfer-learning or using all the pre-trained parameters from ImageNet (Deng et al., 2009). The results showed that training VGG19 freezing only the first six layers and unfreezing the rest, achieved the highest metrics. In respect of the hyperparameters, we used a 32 batch size, binary crossentropy loss, Adam optimizer with 1e-6 learning rate, and relu function in fully connected layers.

In InceptionV3's training, we insisted in doing transfer-learning due to the best results in the previous backbone. Thus, we froze all the layers until layer number 249, and we unfroze the rest. We changed the batch size from 32 in VGG19 training to 128 because InceptionV3 trains faster than VGG19, and used to overfit. We used relu function and binary crossentropy again, but we changed Adam for RMSprop with learning rate 1e-5 because InceptionV3 was designed to be optimal with RMSprop.

In addition, we repeated transfer-learning strategy with MobileNetV2, freezing only the first four layers. We used 64 batch size, binary crossentropy loss and RMSprop optimizer with 1e-5 learning rate.

As we did several training changing dropout values, the number of epochs varies for each dropout values. Nonetheless, the best results were achieved (see Table 1) with 100 epochs for VGG19 backbone, and 10 epochs for InceptionV3 and MobileNetV2.

4.2 Detection Results

In this article, we did two experiments. The first one consisted in training three different models, using the CNN that we mentioned in section 3.1, applying different dropout values. This helps us to know which CNN gets better results and, also, this experiment shows that our models are not overfitting during training. We also measured the average inference time with the same test set, 1644 (10%) samples from all the sensors (A, B, and C) (see Table 1) .

Table 1: Accuracy (Acc), Precision (P), Recall (R) and Average Inference Time (Time) from the test results of the CNN with each backbone.

Backbone	Acc	P	R	Time
$VGG19$	99.9%	99.8%	100%	140ms
$Inception$	99.7%	99.5%	100%	90ms
$MobileNet$	98.1%	97.3%	98.9%	70ms

As can be seen in Table 2, the results are really promising. Everything indicates that it is possible to keep good performance even if we increase the training grasping dataset size looking for a better generalisation of the neural models. Although all the results are similar, VGG19 and InceptionV3 achieve better percentages than MobileNetV2. This happens because MobileNetV2 loses performance in order to achieve an extremely low inference time. Besides, VGG19 is slower than InceptionV3 and MobileNetV2, both in training and inference time.

The second and last experiment we carried out allows us to see how well our models are generalising among sensors. In the previous experiment, we trained the neural models with data from all the sensors (see Figure 3), so we know our models work fine in this case. Now, we want to see if the neural models can generalise from one sensor and detect contact using another sensor. This point is very interesting to help us analyse the model's behaviour in case the sensor unit breaks and has to be replaced by another one - which can be common in low-cost sensors.

Therefore, the idea is to train two models for the sensor units A and B. To do so, model A is trained only with data from sensor A (6186 samples) and model B is trained with data from sensor B (4367). Then, we test model A with a test set from sensor A (687) and another test from sensor B (436). We repeat the process with sensor B. Results can be seen in Figure 5.

Later, we train another model with data from sensors A and B (6186,4367) and we test it with a new test set from sensors A and B. This process is repeated for each CNN (see Figure 6). Finally, a third model is trained with data from sensors A, B, and C. Testing results are shown in Table 1.

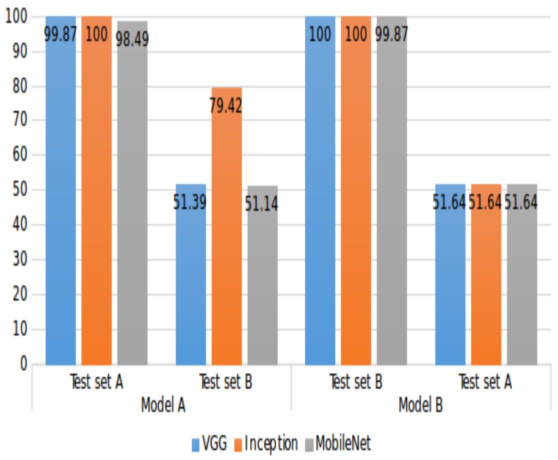

Figure 5: Test results by training model A and B separately.

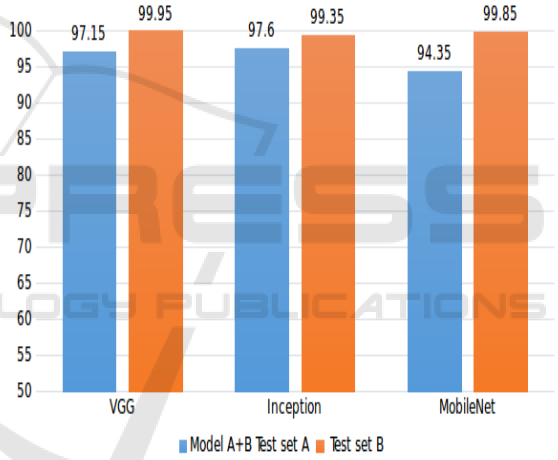

Figure 6: Test results by training model A and B with data from sensors A and B.

As can be seen in Figure 6 and Table 1, adding samples from different units of DIGIT sensor helps the system to generalise better. Thus, it is possible to increase CNN's performance by manufacturing more sensors and adding their images to the training dataset.

Taking into account the results of both experiments, InceptionV3 backbone is the best option to choose. Although VGG19's and InceptionV3's metrics are almost identical, InceptionV3's inference and training time are much better than VGG19's.

Table 2: VGG19, InceptionV3 and MobileNetV2 metrics for each dropout values.

Backbone	Dropout values					
	No dropout	0.3	0.4	0.5	0.6	
VGG	-	99.26	-	99.94	-	Accuracy (%)
	-	98.55	-	99.88	-	Precision (%)
	-	100	-	100	-	Recall (%)
	-	99.27	-	99.94	-	F1 score (%)
Inception	-	-	99.70	-	99.75	Accuracy (%)
	-	-	99.39	-	99.51	Precision (%)
	-	-	100	-	100	Recall (%)
	-	-	99.69	-	99.76	F1 score (%)
MobileNet	98.10	-	-	-	-	Accuracy (%)
	97.35	-	-	-	-	Precision (%)
	98.90	-	-	-	-	Recall (%)
	98.12	-	-	-	-	F1 score (%)

5 CONCLUSIONS AND FUTURE WORKS

In this paper, we proved that high accuracy grip detection task could be achieved using low-cost visual-based tactile sensors, and a CNN. Although it is difficult to generalise among sensors, better generalisation could be achieved through sensor's industrialisation. Nonetheless, we keep working on improving the manual manufacturing, and assembling of DIGIT sensor units. Manufacturing improvements allow sensor images to be more alike, which will improve model generalisation.

Now, we plan to develop a grasp, hold, release and slip detection system using these results and a LSTM neural network. Implementing a robotic grasping controller is our goal (see Figure 7). Results are shown in this video: https://youtu.be/TxoR9Xm1pcI

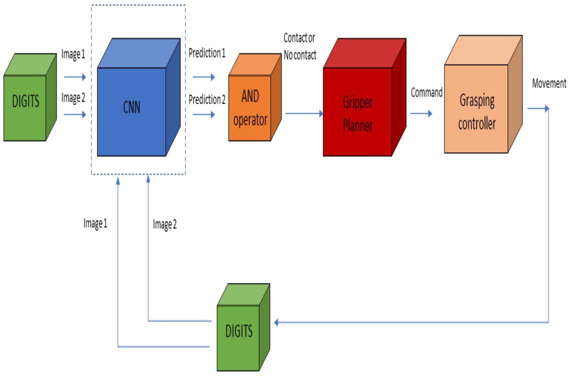

Figure 7: Grasping controller.

ACKNOWLEDGEMENTS

Research work was completely funded by the European Commission and FEDER through the COMMANDIA project (SOE2/P1/F0638), supported by Interreg-V Sudoe. Computer facilities used were provided by Valencia Government and FEDER through the IDIFEDER/2020/003.

REFERENCES

Bekiroglu, Y., Laaksonen, J., Jorgensen, J. A., Kyrki, V., and Kragic, D. (2011). Assessing grasp stability based on learning and haptic data. *IEEE Transactions on Robotics*, 27(3):616–629.

Calandra, R., Owens, A., Jayaraman, D., Lin, J., Yuan, W., Malik, J., Adelson, E. H., and Levine, S. (2018). More than a feeling: Learning to grasp and regrasp using vision and touch. *IEEE Robotics and Automation Letters*, 3(4):3300–3307.

Cockbum, D., Roberge, J.-P., Maslyczyk, A., Duchaine, V., et al. (2017). Grasp stability assessment through unsupervised feature learning of tactile images. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2238–2244. IEEE.

Delgado Rodríguez, Á., Jara, C. A., and Torres, F. (2017). In-hand recognition and manipulation of elastic objects using a servo-tactile control strategy.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Donlon, E., Dong, S., Liu, M., Li, J., Adelson, E., and Rodriguez, A. (2018). Gelslim: A high-resolution, compact, robust, and calibrated tactile-sensing finger. In *2018 IEEE/RSJ International Conference on Intelli-*

gent Robots and Systems (IROS), pages 1927–1934. IEEE.

Garcia-Garcia, A., Zapata-Impata, B. S., Orts-Escolano, S., Gil, P., and Garcia-Rodriguez, J. (2019). Tactilegcn: A graph convolutional network for predicting grasp stability with tactile sensors. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Kappassov, Z., Corrales, J.-A., and Perdereau, V. (2015). Tactile sensing in dexterous robot hands. *Robotics and Autonomous Systems*, 74:195–220.

Kwiatkowski, J., Cockburn, D., and Duchaine, V. (2017). Grasp stability assessment through the fusion of proprioception and tactile signals using convolutional neural networks. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 286–292. IEEE.

Lambeta, M., Chou, P.-W., Tian, S., Yang, B., Maloon, B., Most, V. R., Stroud, D., Santos, R., Byagowi, A., Kammerer, G., et al. (2020). Digit: A novel design for a low-cost compact high-resolution tactile sensor with application to in-hand manipulation. *IEEE Robotics and Automation Letters*, 5(3):3838–3845.

Li, Q., Kroemer, O., Su, Z., Veiga, F. F., Kaboli, M., and Ritter, H. J. (2020). A review of tactile information: Perception and action through touch. *IEEE Transactions on Robotics*, 36(6):1619–1634.

Luo, S., Bimbo, J., Dahiya, R., and Liu, H. (2017). Robotic tactile perception of object properties: A review. *Mechatronics*, 48:54–67.

Ni, P., Zhang, W., Bai, W., Lin, M., and Cao, Q. (2019). A new approach based on two-stream cnns for novel objects grasping in clutter. *J. Intell. Robotic Syst.*, 94(1):161–177.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. in proc. ieee conference on computer vision and pattern recognition.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture. In *European Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yi, Z., Zhang, Y., and Peters, J. (2018). Biomimetic tactile sensors and signal processing with spike trains: A review. *Sensors and Actuators A: Physical*, 269:41–52.

Yuan, W., Dong, S., and Adelson, E. H. (2017). Gelsight: High-resolution robot tactile sensors for estimating geometry and force. *Sensors*, 17(12):2762.

Zapata-Impata, B. S., Gil, P., and Torres, F. (2018). Non-matrix tactile sensors: How can be exploited their local connectivity for predicting grasp stability? *arXiv preprint arXiv:1809.05551*.

Zapata-Impata, B. S., Gil, P., and Torres, F. (2019). Tactile-driven grasp stability and slip prediction. *Robotics*, 8(4):85.